Proxmox VE: klaster trzech węzłów i HA

Poziom: średnio

Dotyczy: Proxmox VE 9.2 (Debian 13 „Trixie”, jądro 7.0), Ceph Squid i Tentacle, ZFS 2.4, laboratorium pve01 do pve03

Czas: ~120 min

Proxmox VE to linuksowy odpowiednik klastra Hyper-V z Failover Cluster: trzy serwery tworzą klaster, wspólny magazyn trzyma dyski maszyn, a menedżer HA uruchamia maszynę na innym węźle, gdy jej węzeł padnie. W Zielonej produkcja stoi na Hyper-V (`hv01`, `hv02`), a Proxmox opisujemy jako laboratorium i jako alternatywę przy wymianie sprzętu. Wracamy tu przy budowie laboratorium, przy awarii węzła i przy rozmowie o tym, czy zmieniać hypervisor.

> Nic z tej strony nie było uruchomione na maszynach Zielonej. Polecenia i parametry sprawdzaliśmy w dokumentacji Proxmoksa (stan na 3 października 2026), skrypty tylko składniowo (`bash -n`), a odpowiedzi poleceń pokazują kształt wyniku, nie zapis z konsoli. Czego nie udało się potwierdzić, oznaczamy słowami „do potwierdzenia”.

> **ZAŁOŻENIA SCENARIUSZA**
>
> Proxmox pracuje jako laboratorium w serwerowni w Pamiątkowie: `pve01` do `pve03` (adresy od 10.20.10.90 do .92), zarządzanie w VLAN 10 przez `sw01` i `sw02`, Corosync w VLAN 101 i Ceph w VLAN 111 bez bramy, BMC w VLAN 98. Produkcja zostaje na Hyper-V (`hv01` i `hv02`).

Skróty użyte na tej stronie (36)

Skrót	Co znaczy
BMC	Baseboard Management Controller: osobny malutki komputer na płycie serwera, który działa nawet przy wyłączonym serwerze i odczytuje m.in. temperatury czujników
CIDR	Classless Inter-Domain Routing: zapis podsieci z długością maski, na przykład 10.20.10.0/24
Corosync	usługa komunikacji klastra, która pilnuje członkostwa węzłów i kworum; Proxmox VE i Pacemaker używają jej do sprawdzania, które węzły żyją

Skrót	Co znaczy
CRS	Cluster Resource Scheduling: harmonogram zasobów klastra Proxmoksa, który wybiera węzeł dla maszyny pod opieką HA w trybie basic, static albo dynamic
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
GB	Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej
GiB	gibibajt: 1024 MiB, czyli około 1,07 GB; tej jednostki używa dokumentacja MailCow przy wymaganiach pamięci
HA	High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu
HBA	Host Bus Adapter: karta, która tylko wystawia dyski systemowi, bez składania ich w macierz; wymagana w trybie przezroczystym dla S2D
IPMI	Intelligent Platform Management Interface: standard, przez który administrator pyta kontroler BMC serwera o temperatury, napięcia i obroty wentylatorów
iSCSI	Internet Small Computer System Interface: sposób udostępniania dysku z macierzy po zwykłej sieci IP, tak że serwer widzi go jak własny dysk
ISO	International Organization for Standardization: organizacja, która wydaje normy techniczne używane na całym świecie
KVM	Keyboard, Video, Mouse: konsola albo przełącznik, który daje jedną klawiaturę, ekran i mysz do wielu serwerów (w Linuksie KVM to też technika wirtualizacji)
LACP	Link Aggregation Control Protocol: protokół z normy IEEE 802.1AX, w którym dwa urządzenia uzgadniają łączenie kilku kabli w jedno łącze logiczne; jedna sesja zostaje zawsze na jednym kablu
LXC	Linux Containers: wczesny mechanizm izolacji procesów w jądrze Linuksa, na którym pierwotnie oparto Dockera
MC-LAG	Multi-Chassis Link Aggregation Group: agregacja łączy do dwóch osobnych przełączników, która działa jak jeden przełącznik dla serwera
NFS	Network File System: udostępnianie katalogów po sieci w świecie Linuksa
OSD	Object Storage Daemon: demon Ceph obsługujący jeden dysk z danymi
PCI	Peripheral Component Interconnect: magistrala kart w serwerze; maszyna z przekazaną kartą PCI jest przypięta do węzła i nie migruje na żywo
PDU	Power Distribution Unit: listwa zasilająca w szafie rack, w droższych wersjach z gniazdami mierzonymi i wejściem na czujnik temperatury
QDevice	Quorum Device: trzeci głos dla klastra z dwoma węzłami, działający na osobnym hoście; w Proxmox VE i Pacemakerze rozstrzyga remis
QEMU	Quick Emulator: program, który uruchamia maszyny wirtualne na Linuksie, z KVM jako akceleracją sprzętową; Proxmox VE zarządza maszynami przez niego
RAID	Redundant Array of Independent Disks: sposób łączenia kilku fizycznych dysków w jeden, widziany przez system jako pojedynczy dysk, zwykle z dodatkową odpornością na awarię jednego z nich

Skrót	Co znaczy
RBD	RADOS Block Device: urządzenie blokowe udostępniane przez Ceph
REST	sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu
S2D	Storage Spaces Direct: programowa macierz Windows Server z dysków wielu serwerów
SET	Switch Embedded Teaming: zespół kart sieciowych zbudowany w przełączniku wirtualnym Hyper-V; łączy dwie karty w jedno łącze bez ustawień po stronie przełącznika fizycznego, więc bez LACP
SSD	Solid State Drive: dysk półprzewodnikowy
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera
STONITH	Shoot The Other Node In The Head: odcięcie węzła, co do którego nie mamy pewności, że jest martwy, najczęściej przez wyłączenie jego zasilania
TCP	Transmission Control Protocol: protokół sieciowy z potwierdzaniem dostarczenia
UDP	User Datagram Protocol: szybki protokół sieciowy bez potwierdzeń
VLAN	Virtual LAN: wydzielona logicznie sieć na tym samym sprzęcie
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS
XFS	system plików stworzony przez Silicon Graphics, domyślny w Rocky Linux i RHEL
ZFS	system plików z migawkami i replikacją przyrostową zbudowany w jednym narzędziu z menedżerem wolumenów (dawniej Zettabyte File System)

Fiszka

Komendy wykonujemy jako root w powłoce węzła, przez SSH albo konsolę w interfejsie WWW (na Proxmoksie domyślnie nie ma `sudo`). Adresy z Zielonej: `pve01` to `10.20.10.90`, `pve02` to `10.20.10.91`, `pve03` to `10.20.10.92`, a sieć Corosync (link0) to `10.20.101.0/24` z tą samą końcówką. Nazwy `NazwaKlastra`, `AdresLink0`, `NumerMaszyny` i podobne zastępujemy własnymi.

TYLKO ODCZYT

Sprawdzenie wersji Proxmoksa: pveversion

```
pveversion
```

Pokazuje wersję menedżera Proxmoksa i działające jądro. Porównujemy ją na wszystkich węzłach, bo klaster z różnymi wersjami zostawiamy tylko na czas aktualizacji.

Przykład (kształt odpowiedzi):

```
pve-manager/9.2.x/xxxxxxxxxxxxxxxx (running kernel: 7.0.x-x-pve)
```

□□ **pveversion** = pokazuje wersję Proxmox VE i pakietów, z których składa się instalacja

□□ ZMIENIA SYSTEM

Utworzenie klastra: pvecm create

```
pvecm create NazwaKlastra --link0 AdresLink0,priority=20 --link1 AdresLink1,priority=10
```

> „ (gdzie `AdresLink0` to adres węzła w dedykowanej sieci Corosync, a `AdresLink1` to jego adres w sieci zarządzania; wyższy priorytet oznacza łącze używane w pierwszej kolejności [2])

Wykonujemy tylko na pierwszym węźle (`pve01`) i tylko raz. Nazwy klastra ani adresów węzłów nie zmieniamy później, więc ustalamy je przed tym poleceniem [2].

□□ **Corosync** = usługa komunikacji klastra, która pilnuje członkostwa węzłów i kworum; Proxmox VE i Pacemaker używają jej do sprawdzania, które węzły żyją · **pvecm** = Proxmox VE Cluster Manager: tworzy klaster, dołącza węzły i pokazuje kworum, na przykład pvecm status

□□ ZMIENIA SYSTEM

Dołączenie węzła do klastra: pvecm add

```
pvecm add AdresPierwszegoWezla --link0 AdresLink0TegoWezla --link1 AdresLink1TegoWezla
```

> „ (gdzie `AdresPierwszegoWezla` to adres `pve01`, a pozostałe dwa to adresy węzła, który dołączamy, w tych samych dwóch sieciach)

Wykonujemy na `pve02` i `pve03`, po jednym. Polecenie pyta o hasło root pierwszego węzła i o odcisk jego certyfikatu, a dołączany węzeł nie może mieć żadnych maszyn, bo jego konfiguracja zostanie zastąpiona konfiguracją klastra [2].

□□ **pvecm** = Proxmox VE Cluster Manager: tworzy klaster, dołącza węzły i pokazuje kworum, na przykład pvecm status

□□ TYLKO ODCZYT

Sprawdzenie kworum: pvecm status

```
pvecm status
```

Pokazuje liczbę węzłów, oczekiwane i obecne głosy oraz to, czy klaster ma kworum. Szukamy `Quorate: Yes` i `Total votes` równego liczbie węzłów.

Przykład (kształt odpowiedzi, `pve01`):

```
Cluster information
-------------------
Name:             zielona-pve
Config Version:   3
Transport:        knet
Secure auth:      on

Quorum information
------------------
Nodes:            3
Quorate:          Yes

Votequorum information
----------------------
Expected votes:   3
Highest expected: 3
Total votes:      3
Quorum:           2
Flags:            Quorate
```

pvecm = Proxmox VE Cluster Manager: tworzy klaster, dołącza węzły i pokazuje kworum, na przykład pvecm status

TYLKO ODCZYT

Lista węzłów klastra: pvecm nodes

```
pvecm nodes
```

Wypisuje węzły z numerem, liczbą głosów i nazwą. Brak któregoś węzła na liście oznacza, że nie widzi go węzeł, na którym pytamy.

pvecm = Proxmox VE Cluster Manager: tworzy klaster, dołącza węzły i pokazuje kworum, na przykład pvecm status

TYLKO ODCZYT

Sprawdzenie łączy Corosync: corosync-cfgtool -s

```
corosync-cfgtool -s
```

Pokazuje stan każdego łącza (link0 i link1) do każdego węzła. Przy zdrowym klastrze oba łącza do każdego węzła są połączone, a rozłączone link0 przy działającym link1 jest ostrzeżeniem, że straciliśmy redundancję.

Corosync = usługa komunikacji klastra, która pilnuje członkostwa węzłów i kworum; Proxmox VE i Pacemaker używają jej do sprawdzania, które węzły żyją · **corosync-cfgtool** = Narzędzie Corosync do podglądu stanu łączy między węzłami; polecenie corosync-cfgtool -s pokazuje, czy każdy link działa

ZMIENIA SYSTEM

Dodanie urządzenia kworum dla dwóch węzłów: pvecm qdevice setup

```
pvecm qdevice setup AdresQDevice
```

> (gdzie `AdresQDevice` to adres hosta spoza klastra z zainstalowanym `corosync-qnetd`; na hoście ma być dostęp root przez klucz SSH albo tymczasowo przez hasło [2])

Dajemy dwuwęzłowemu klastrowi trzeci głos. Pakiety i szczegóły opisujemy w sekcji o kworum, a w klastrze z trzema węzłami urządzenia kworum nie dokładamy [2].

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera · **pvecm** = Proxmox VE Cluster Manager: tworzy klaster, dołącza węzły i pokazuje kworum, na przykład pvecm status

OSTROŻNIE · NIEODWRACALNE

Awaryjne obniżenie oczekiwanych głosów: pvecm expected 1

```
pvecm expected 1
```

Pozwala węzłowi bez kworum działać tak, jakby wystarczał mu jeden głos. Robimy to tylko wtedy, gdy pozostałe węzły na pewno są wyłączone, bo inaczej dwie połówki klastra mogą zapisywać niezależnie [2].

pvecm = Proxmox VE Cluster Manager: tworzy klaster, dołącza węzły i pokazuje kworum, na przykład pvecm status

ZMIENIA SYSTEM

Przeładowanie sieci po zmianie konfiguracji: ifreload -a

```
ifreload -a
```

Stosuje zmiany z `/etc/network/interfaces` bez restartu węzła (pakiet `ifupdown2` jest w Proxmoksie domyślnie) [7]. Zmiany sieci wykonujemy z konsoli BMC albo w oknie serwisowym, bo błąd może odciąć nas od węzła.

BMC = Baseboard Management Controller: osobny malutki komputer na płycie serwera, który działa nawet przy wyłączonym serwerze i odczytuje m.in. temperatury czujników ·**ifreload** = ifupdown2: stosuje zmiany z /etc/network/interfaces bez restartu węzła, ifreload -a

ZMIENIA SYSTEM

Przygotowanie Ceph na węzłach: pveceph install

```
pveceph install
```

Instaluje pakiety Ceph na węźle. Wersję (Squid albo Tentacle) wybieramy w kreatorze w interfejsie WWW albo opcją polecenia, którą pokazuje `pveceph install --help` [4].

WWW = World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS · **pveceph** = Proxmox VE Ceph: instaluje i konfiguruje Ceph w klastrze Proxmoksa (monitory, dyski OSD, pule)

ZMIENIA SYSTEM

Pierwsza konfiguracja sieci Ceph: pveceph init

```
pveceph init --network SiecCeph
```

> (gdzie `SiecCeph` to podsieć Ceph w zapisie CIDR, u nas `10.20.111.0/24`; osobną sieć wewnętrzną dla replikacji dodaje opcja `--cluster-network` [4])

Wykonujemy jeden raz na dowolnym węźle klastra. Tworzy minimalny plik `/etc/pve/ceph.conf`, a reszta ustawień leży w bazie monitorów.

CIDR = Classless Inter-Domain Routing: zapis podsieci z długością maski, na przykład 10.20.10.0/24 · **pveceph** = Proxmox VE Ceph: instaluje i konfiguruje Ceph w klastrze Proxmoksa (monitory, dyski OSD, pule)

ZMIENIA SYSTEM

Utworzenie monitora i menedżera Ceph: pveceph mon create

```
pveceph mon create
```

Wykonujemy na każdym z trzech węzłów, a zaraz potem `pveceph mgr create`, żeby każdy węzeł z monitorem miał też menedżera. Trzy monitory dają kworum Ceph przy awarii jednego węzła [4].

pveceph = Proxmox VE Ceph: instaluje i konfiguruje Ceph w klastrze Proxmoksa (monitory, dyski OSD, pule)

OSTROŻNIE · NIEODWRACALNE

Utworzenie OSD na dysku: pveceph osd create

```
pveceph osd create /dev/sdX
```

> (gdzie `/dev/sdX` to dysk pod OSD; dysk jest czyszczony, więc sprawdzamy jego nazwę poleceniem `lsblk`, zanim wciśniemy Enter)

Jeden dysk to jeden OSD, a dysk ma być widoczny dla systemu wprost, bez macierzy sprzętowej [4]. Zaczynamy od dwóch dysków na węzeł, żeby po awarii jednego OSD było gdzie odtworzyć dane.

OSD = Object Storage Daemon: demon Ceph obsługujący jeden dysk z danymi · **pveceph** = Proxmox VE Ceph: instaluje i konfiguruje Ceph w klastrze Proxmoksa (monitory, dyski OSD, pule)

◫◫ ZMIENIA SYSTEM

Utworzenie puli Ceph dla maszyn: pveceph pool create

```
pveceph pool create NazwaPuli --add_storages
```

Tworzy pulę replikowaną z domyślnym `size` 3 i `min_size` 2 i od razu dodaje ją jako magazyn Proxmoksa [4]. Wartości `min_size` równej 1 nie ustawiamy nigdy, bo dokumentacja ostrzega, że grozi to utratą danych [4].

◫◫ **pveceph** = Proxmox VE Ceph: instaluje i konfiguruje Ceph w klastrze Proxmoksa (monitory, dyski OSD, pule)

◫◫ TYLKO ODCZYT

Sprawdzenie stanu Ceph: ceph -s

```
ceph -s
```

Pokazuje zdrowie klastra (`HEALTH_OK`, `HEALTH_WARN`, `HEALTH_ERR`), monitory w kworum, OSD i grupy umieszczania. Przy zdrowym klastrze wszystkie grupy mają stan `active+clean`.

Przykład (kształt odpowiedzi):

```
  cluster:
    health: HEALTH_OK

  services:
    mon: 3 daemons, quorum pve01,pve02,pve03
    mgr: pve01(active), standbys: pve02, pve03
    osd: 6 osds: 6 up, 6 in

  data:
    pools:   1 pools, 128 pgs
    pgs:     128 active+clean
```

◫◫ **OSD** = Object Storage Daemon: demon Ceph obsługujący jeden dysk z danymi · **ceph** = administracja klastrem Ceph, na przykład ceph -s pokazuje jego stan

◫◫ ZMIENIA SYSTEM

Utworzenie zadania replikacji ZFS: pvesr create-local-job

```
pvesr create-local-job NumerMaszyny-0 NazwaWezlaDocelowego --schedule "*/15"
```

> (gdzie `NumerMaszyny-0` to identyfikator zadania: numer maszyny i numer kolejny zadania, na przykład `100-0`, a `NazwaWezlaDocelowego` to węzeł, na który kopiujemy dyski)

Replikuje dyski maszyny ze ZFS na ZFS drugiego węzła co 15 minut. Magazyn o tej samej nazwie musi istnieć na obu węzłach, a po awarii tracimy zmiany od ostatniej udanej replikacji [5].

ZFS = system plików z migawkami i replikacją przyrostową zbudowany w jednym narzędziu z menedżerem wolumenów (dawniej Zettabyte File System) · **pvesr** = Proxmox VE Storage Replication: zadania replikacji dysków ZFS między węzłami, domyślnie co 15 minut

TYLKO ODCZYT

Sprawdzenie stanu replikacji: pvesr status

```
pvesr status
```

Pokazuje zadania replikacji z tego węzła, czas ostatniej synchronizacji, czas następnej i liczbę błędów. Zadanie z rosnącą liczbą błędów powtarza próbę co 30 minut, więc sprawdzamy jego przyczynę, zamiast czekać [5].

pvesr = Proxmox VE Storage Replication: zadania replikacji dysków ZFS między węzłami, domyślnie co 15 minut

ZMIENIA SYSTEM

Dodanie maszyny do HA: ha-manager add

```
ha-manager add vm:NumerMaszyny --state started --max_restart 1 --max_relocate 1
```

> (gdzie `vm:NumerMaszyny` to identyfikator zasobu, na przykład `vm:100`; dla kontenera piszemy `ct:NumerKontenera`)

Oddaje maszynę pod opiekę menedżera HA. Wartości `max_restart` i `max_relocate` to domyślne jedynki: jedna próba restartu na tym samym węźle i jedna próba przeniesienia na inny, a potem zasób przechodzi w stan `error` [3][6].

HA = High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu · **ha-manager** = menedżer wysokiej dostępności Proxmox VE: dodaje maszynę do zasobów HA (add), ustawia jej stan (set) i pokazuje stan klastra (status)

TYLKO ODCZYT

Sprawdzenie stanu HA: ha-manager status

```
ha-manager status
```

Pokazuje, czy klaster ma kworum, który węzeł jest menedżerem (master), stan lokalnych menedżerów każdego węzła i każdy zasób z jego węzłem i stanem.

Przykład (kształt odpowiedzi):

```
quorum OK
master pve01 (active, Fri Oct  2 16:05:12 2026)
lrm pve01 (idle, Fri Oct  2 16:05:15 2026)
lrm pve02 (active, Fri Oct  2 16:05:14 2026)
lrm pve03 (idle, Fri Oct  2 16:05:15 2026)
service vm:100 (pve02, started)
```

HA = High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu · **ha-manager** = menedżer wysokiej dostępności Proxmox VE: dodaje maszynę do zasobów HA (add), ustawia jej stan (set) i pokazuje stan klastra (status)

ZMIENIA SYSTEM

Wskazanie preferowanych węzłów maszyny: ha-manager rules add node-affinity

```
ha-manager rules add node-affinity NazwaReguly --resources vm:NumerMaszyny --nodes
pve01:2,pve02:1,pve03:1
```

Tworzy regułę powinowactwa do węzłów: maszyna woli `pve01` (priorytet 2), a w razie jego awarii działa na `pve02` lub `pve03`. Dopisanie `--strict 1` zabrania uruchamiania maszyny poza wymienionymi węzłami [3]. Reguły zastąpiły grupy HA od Proxmoksa 9.0 [3].

HA = High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu · **ha-manager** = menedżer wysokiej dostępności Proxmox VE: dodaje maszynę do zasobów HA (add), ustawia jej stan (set) i pokazuje stan klastra (status)

ZMIENIA SYSTEM

Włączenie trybu konserwacji węzła: ha-manager crm-command node-maintenance

```
ha-manager crm-command node-maintenance enable NazwaWezla
```

Menedżer HA przenosi z węzła wszystkie zasoby HA i uznaje go za niedostępny. Po pracach wykonujemy to samo polecenie z `disable`, a zasoby wracają na węzeł, z którego je zabrano [3].

HA = High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu · **ha-manager** = menedżer wysokiej dostępności Proxmox VE: dodaje maszynę do zasobów HA (add), ustawia jej stan (set) i pokazuje stan klastra (status)

ZMIENIA SYSTEM

Migracja działającej maszyny na inny węzeł: qm migrate --online

```
qm migrate NumerMaszyny NazwaWezlaDocelowego --online
```

Przenosi uruchomioną maszynę bez wyłączania jej, o ile nie ma urządzeń przypiętych do węzła (na przykład przekazanej karty PCI) [11]. Maszynę pod opieką HA przenosimy raczej poleceniem `ha-manager migrate`, żeby menedżer HA wiedział o zmianie.

PCI = Peripheral Component Interconnect: magistrala kart w serwerze; maszyna z przekazaną kartą PCI jest przypięta do węzła i nie migruje na żywo · **HA** = High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu

ZMIENIA SYSTEM

Migracja wyłączonej maszyny z dyskami lokalnymi: qm migrate --with-local-disks

```
qm migrate NumerMaszyny NazwaWezlaDocelowego --with-local-disks
```

Dla maszyn na lokalnym magazynie (na przykład ZFS bez Ceph) kopiuje także dyski [11]. Czas zależy od rozmiaru dysków, a przy włączonej replikacji ZFS kopiowana jest tylko różnica

od ostatniej synchronizacji [5].

□□ **ZFS** = system plików z migawkami i replikacją przyrostową zbudowany w jednym narzędziu z menedżerem wolumenów (dawniej Zettabyte File System)

□□ TYLKO ODCZYT

Sprawdzenie usługi watchdoga: systemctl status watchdog-mux

```
systemctl status watchdog-mux
```

Pokazuje, czy działa usługa, która udostępnia watchdog menedżerowi HA i którego modułu używa. Bez działającego watchdoga węzeł nie ma jak się sam wyłączyć, więc fencing nie zadziała [3].

□□ **HA** = High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu · **systemctl** = system control: sterowanie usługami systemd

□□ PLIK KONFIGURACYJNY

Włączenie sprzętowego watchdoga: /etc/default/pve-ha-manager

```
WATCHDOG_MODULE=iTCO_wdt
```

Domyślnie wszystkie moduły sprzętowego watchdoga są zablokowane, a bez wskazanego modułu Proxmox używa programowego `softdog` [3]. Nazwa `iTCO_wdt` to przykład z dokumentacji dla układów Intela, a moduł dla naszych serwerów sprawdzamy w opisie płyty i BMC (do potwierdzenia).

□□ **BMC** = Baseboard Management Controller: osobny malutki komputer na płycie serwera, który działa nawet przy wyłączonym serwerze i odczytuje m.in. temperatury czujników

□□ OSTROŻNIE · NIEODWRACALNE

Wyłączenie węzła przez BMC w teście awarii: ipmitool chassis power off

```
export IPMI_PASSWORD='HasloKontaBMC'
ipmitool -I lanplus -H AdresBMC -U NazwaUsera -E chassis power off
```

(gdzie `AdresBMC` to adres BMC węzła w VLAN 98, na przykład `10.20.98.91` dla `pve02`; opcja `-E` czyta hasło ze zmiennej, jak na stronie o BMC)

Twarde odcięcie zasilania węzła odtwarza awarię prądu lepiej niż zwykłe wyłączenie z systemu. Wykonujemy to wyłącznie w laboratorium albo w oknie testowym, ze świadomością, że maszyny z tego węzła staną.

BMC = Baseboard Management Controller: osobny malutki komputer na płycie serwera, który działa nawet przy wyłączonym serwerze i odczytuje m.in. temperatury czujników · **VLAN** = Virtual LAN: wydzielona logicznie sieć na tym samym sprzęcie · **export** = ustaw zmienną środowiskową widoczną dla uruchamianych programów · **ipmitool** = Intelligent Platform Management Interface tool: program, który przez IPMI pyta kontroler BMC serwera o czujniki, np. sdr type temperature wypisuje temperatury

Z życia wzięte

Z ŻYCIA WZIĘTE

Środa, 14:10. Pan Zbigniew pyta przy kawie, co firma zrobi, gdy skończy się gwarancja na `hv01` i `hv02`, i czy Windows Server jest jedyną drogą. Ola wyciąga z szafy trzy stare serwery i w godzinę robi z nich `pve01`, `pve02` i `pve03`, a Tomek przez BMC wyłącza zasilanie `pve02` z działającą maszyną `ci-test`. Pan Zbigniew patrzy na ekran, na którym przez dwie minuty nie dzieje się nic, a potem maszyna wstaje na innym węźle i Tomek notuje, że przerwa była dokładnie taka, jak obiecuje dokumentacja, czyli rzędu dwóch minut.

Ta przerwa jest ceną automatycznego przełączenia: klaster musi najpierw uznać węzeł za martwy, a dopiero potem wolno mu uruchomić jego maszyny gdzie indziej. Na tej stronie budujemy takie laboratorium od instalacji po test awarii, a o różnicach względem Hyper-V piszemy na stronie Proxmox VE a Hyper-V z Failover Cluster: porównanie i przenosiny.

Jak to działa

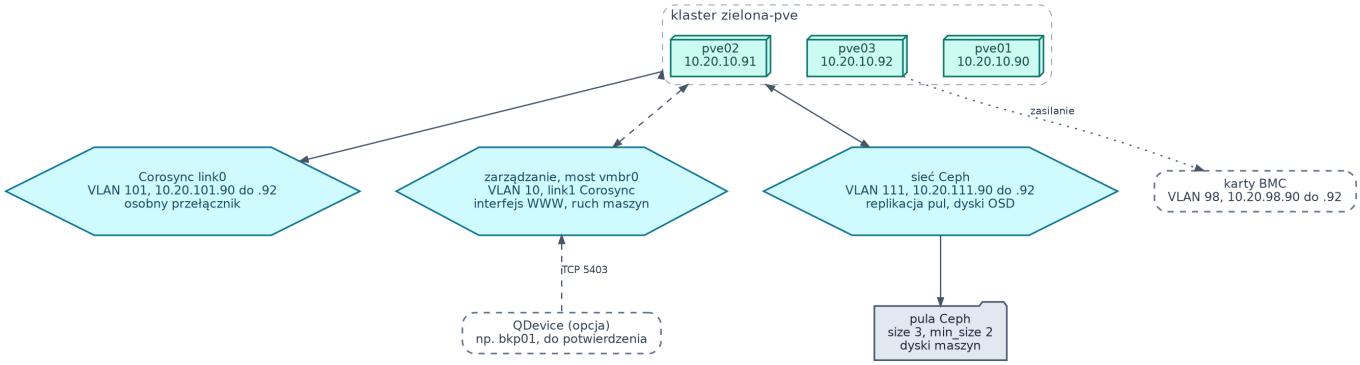
Rozwiń: jak to działa pod spodem

Z czego składa się klaster

Pojęcie: Proxmox VE. Linuksowy system (nakładka na Debiana) do uruchamiania maszyn wirtualnych (KVM i QEMU) oraz kontenerów (LXC), z zarządzaniem przez przeglądarkę, wiersz poleceń i interfejs REST. Kod jest otwarty (licencja AGPLv3), a płatna jest tylko opcjonalna subskrypcja ze wsparciem [10].

Pojęcie: klaster Proxmoksa. Grupa węzłów, które mają wspólną konfigurację w katalogu `/etc/pve`, kopiowanym na każdy węzeł i zapisywanym tylko wtedy, gdy klaster ma kworum. Nie ma osobnego serwera zarządzania: interfejs WWW działa na każdym węźle i pokazuje cały klaster, więc awaria jednego węzła nie odcina nas od reszty.

Klaster Zielonej składa się z trzech węzłów, trzech sieci i wspólnego magazynu. Rysunek pokazuje, co z czym rozmawia i dlaczego sieci są oddzielne: Corosync dostaje własny kabel i własny VLAN, Ceph własne łącza 10G, a BMC leży w sieci zarządzania poza serwerami.



Rysunek 1. Laboratorium Proxmox VE w Zielonej. Trzy węzły z adresami 10.20.10.90 do 10.20.10.92 łączą trzy osobne sieci: Corosync (VLAN 101, link0), zarządzania z mostem maszyn (VLAN 10, link1) i Ceph (VLAN 111). Dyski maszyn leżą w puli Ceph, a karty BMC w VLAN 98 mają te same końcówki adresów. (kliknięcie otwiera pełny rozmiar)

Proxmox używa do komunikacji w klastrze programu Corosync, tego samego, na którym działa Pacemaker ze strony Pacemaker na Linuksie: własna usługa z adresem wirtualnym. Corosync odpowiada za członkostwo węzłów, głosowanie i kworum, a nad nim stoi menedżer HA, który decyduje, gdzie ma działać każda maszyna.

Sieci laboratorium

Corosync jest wrażliwy na opóźnienia, a nie na przepustowość: dokumentacja wymaga opóźnień poniżej 5 milisekund między węzłami i radzi, żeby nie dzielić sieci Corosync z

magazynem ani z ruchem maszyn [2]. Dlatego proponujemy trzy sieci, wszystkie z tą samą końcówką adresu co węzeł (`.90` dla `pve01`, `.91` dla `pve02`, `.92` dla `pve03`).

Sieć	VLAN i podsieć	Adres `pve01`	Do czego	Fizycznie
Corosync, link0	VLAN 101, `10.20.101.0/24`, bez bramy	`10.20.101.90`	komunikacja klastra, kworum, wykrywanie awarii	osobny port 1 Gb na każdym węźle, osobny przełącznik
Zarządzanie i maszyny, link1	VLAN 10, `10.20.10.0/24`	`10.20.10.90`	interfejs WWW, SSH, ruch maszyn przez most `vmbr0`, zapasowe łącze Corosync	dwa łącza 10G w `bond0` (active-backup) do `sw01` i `sw02`
Ceph	VLAN 111, `10.20.111.0/24`, bez bramy	`10.20.111.90`	replikacja i odczyt dysków maszyn	te same łącza 10G co wyżej, osobny VLAN
BMC	VLAN 98, `10.20.98.0/24`	`10.20.98.90`	konsola, zasilanie, odczyt czujników	osobny port BMC, zasady ze strony o BMC

Dokumentacja zaleca, żeby każde łącze Corosync miało własne fizyczne połączenie [2], dlatego link1 w naszej propozycji to sieć zarządzania, a nie drugi VLAN na tych samych kablach. Jeszcze lepszy byłby drugi dedykowany port 1 Gb na osobnym przełączniku, jeśli serwery go mają. Podsieci VLAN 101 i VLAN 111 są propozycją do zatwierdzenia przez autora bazy: w świecie Zielonej nie było ich dotąd, a w całej bazie nie znaleźliśmy ich zajętych.

Most `vmbr0` ustawiamy jako świadomy VLAN-ów (`bridge-vlan-aware yes`), żeby maszyny dostawały sieć z dowolnego VLAN-u przez znacznik na ich karcie wirtualnej, bez osobnego mostu na każdą sieć [7]. W trunku do przełączników rdzeniowych dopuszczamy VLAN-y 10, 98, 111 i te, które mają widzieć maszyny, zgodnie z zasadami ze strony Przełączniki: L2 i L3, tablica MAC, VLAN, LACP i STP.

Agregację dwóch łączy robimy w trybie active-backup, a nie LACP: dokumentacja zaleca LACP albo active-backup [7], a LACP do dwóch przełączników wymaga stosu lub MC-LAG, co na stronie o przełącznikach jest tylko propozycją. Active-backup działa na zwykłych trunkach, tak jak zespół SET (Switch Embedded Teaming) na hostach Hyper-V.

Kworum i urządzenie kworum

> **Pojęcie: kworum.** Zasada większości: klaster zapisuje konfigurację i uruchamia maszyny tylko wtedy, gdy widzi więcej niż połowę głosów. Zasadę i tabelę dla 1 do 5 węzłów opisuje strona

> Wysoka dostępność: pojęcia, liczenie dostępności i poziomy rezerwy.

Trzy węzły po jednym głosie dają większość z dwóch, więc klaster przeżywa awarię jednego węzła. Dokumentacja HA wymaga co najmniej trzech węzłów dla wiarygodnego kworum [3], a dwa węzły bez trzeciego głosu stają przy awarii któregokolwiek z nich.

> **Pojęcie: QDevice.** Urządzenie kworum: demon `corosync-qnetd` na hoście spoza klastra, który daje klastrowi dodatkowy głos. Proxmox wspiera je w klastrach z parzystą liczbą węzłów i zaleca dla klastrów dwuwęzłowych, a w nieparzystych odradza, bo QDevice daje wtedy N minus 1 głosów i staje się punktem krytycznym [2].

Dla dwóch węzłów konfigurujemy je tak: na hoście zewnętrznym instalujemy `corosync-qnetd`, na obu węzłach `corosync-qdevice`, a potem na jednym z węzłów wykonujemy `pvecm qdevice setup AdresQDevice` [2]. Port TCP 5403 hosta zewnętrznego musi być osiągalny z węzłów, a stan sprawdzamy w `pvecm status`, gdzie widać linię z `Qdevice` [2]. Kandydatem w Zielonej jest `bkp01` (`10.20.10.61`), który jest już trzecim głosem klastra z Pacemakerem. To, czy jeden `qnetd` obsłuży kilka klastrów i czy `pvecm qdevice setup` działa z hostem Rocky Linux (skrypt zakłada konfigurację Debiana), jest do potwierdzenia, a bezpieczniejszą drogą jest mała maszyna z Debianem.

Magazyn: Ceph albo ZFS z replikacją

Maszyna, którą HA ma uruchomić na innym węźle, potrzebuje swoich dysków na tym węźle. Dokumentacja wymienia kilka rodzajów magazynu współdzielonego (Ceph, NFS, iSCSI), a poza nimi działa replikacja ZFS [9]. Wybór różni się tym, ile danych możemy stracić.

Cecha	Ceph (RBD)	ZFS z replikacją (`pvesr`)	NFS lub iSCSI z zewnątrz
Magazyn współdzielony	tak, dane są na wszystkich węzłach	nie, każdy węzeł ma własną pulę ZFS	tak, ale serwer magazynu to nowy punkt awarii
Utrata danych przy awarii węzła	brak (zapis potwierdzany po trzech kopiach)	zmiany od ostatniej replikacji, domyślnie do 15 minut [5]	zależy od magazynu
Minimum	trzy węzły z dyskami, sieć 10 Gb [4]	dwa węzły (z urządzeniem kworum), zwykła sieć	serwer magazynu
Pojemność użyteczna	około jednej trzeciej surowej przy trzech kopiach	cała pula węzła, ale dane są na dwóch węzłach	zależy od magazynu

Cecha	Ceph (RBD)	ZFS z replikacją (`pvesr`)	NFS lub iSCSI z zewnątrz
Złożoność	wysoka: monitory, OSD, pule, zdrowie klastra	niska: ZFS i zadanie replikacji	niska po stronie Proxmoksa
Dla Zielonej	laboratorium z trzema węzłami	oddział z dwoma węzłami i `bkp01` jako trzecim głosem	`bkp01` odpada, bo to jeden serwer

> **Pojęcie: Ceph i OSD.** Ceph to rozproszony magazyn, który sam rozkłada dane i odtwarza brakujące kopie. Dysk maszyny leży w nim jako obraz blokowy (RBD), a każdy fizyczny dysk obsługuje osobny demon OSD [4].

Pula Ceph ma dwa parametry. `size` to liczba kopii każdego obiektu (domyślnie 3), a `min_size` to najmniejsza liczba kopii, przy której Ceph jeszcze przyjmuje zapisy (domyślnie 2) [4]. Przy trzech węzłach i kopiach rozmieszczanych po jednej na węzeł awaria jednego węzła zostawia dwie kopie: dane są dostępne, zapisy idą dalej, ale klaster pokazuje `HEALTH_WARN`, bo nie ma gdzie odtworzyć trzeciej kopii. Awaria drugiego węzła zostawia jedną kopię, poniżej `min_size`, więc operacje na dyskach maszyn stają.

Dokumentacja wymaga co najmniej trzech podobnych serwerów, sieci 10 Gb dla ruchu Ceph, jednego OSD na dysk i dysków widocznych wprost, bez kontrolera RAID [4]. To ta sama zasada, którą dla S2D opisuje strona RAID i kontrolery: sprzętowy, programowy i HBA dla S2D: system na małym mirrorze, a dyski danych za kontrolerem w trybie HBA. Dla OSD dokumentacja podaje domyślnie 4 GiB pamięci, a dla dobrej wydajności co najmniej 8 GiB na OSD [4]. Wersję Ceph na nowy klaster wybieramy z uwzględnieniem tego, że Squid według planu wydań kończy wsparcie 31 października 2026, o czym pisaliśmy na stronie Replikacja danych między hostami: DRBD, GlusterFS, Ceph, ZFS i replikacja w aplikacji, więc w laboratorium bierzemy Tentacle.

> **Pojęcie: replikacja magazynu (pvesr).** Okresowe kopiowanie dysków maszyny ze ZFS jednego węzła na ZFS drugiego, przez migawki. Domyślny harmonogram to co 15 minut, najkrótszy odstęp to minuta, a najdłuższy raz w tygodniu [5].

HA działa z replikacją, ale dokumentacja mówi wprost, że przy awarii węzła możemy stracić dane między ostatnią synchronizacją a awarią [5]. Gdy maszyna przenosi się na węzeł docelowy, kierunek replikacji odwraca się sam, a przeniesienie na węzeł bez kopii oznacza pełne przesłanie dysków [5]. To zasadniczo ta sama ciepła rezerwa, o której piszemy w książce od początku, tylko z automatycznym uruchomieniem.

HA manager: zasoby, stany i reguły

> **Pojęcie: zasób HA.** Maszyna albo kontener oddany pod opiekę menedżera HA. Zasób ma zawsze wskazany stan, który menedżer stara się utrzymać, a zapisujemy go jako `vm:NumerMaszyny` albo `ct:NumerKontenera` [3].

Menedżer HA składa się z dwóch rodzajów procesów. Menedżer klastra (`pve-ha-crm`) działa aktywnie na jednym węźle i podejmuje decyzje, a menedżer lokalny (`pve-ha-lrm`) działa na każdym węźle i wykonuje polecenia: uruchamia, zatrzymuje i przenosi maszyny. Pod spodem pracuje watchdog, który pilnuje, żeby węzeł bez kworum nie zostawił po sobie działających maszyn.

Stan zasobu	Co oznacza według dokumentacji	Kiedy go używamy
`started`	zasób ma działać, menedżer lokalny uruchamia go jak najszybciej [3]	maszyny produkcyjne, domyślny stan po `ha-manager add`
`stopped`	zasób jest zatrzymany, a menedżer lokalny to potwierdził [3]	maszyna wyłączona celowo, ale nadal pod opieką HA
`disabled`	zasób jest zatrzymany i oznaczony jako wyłączony [3]	maszyna, którą HA ma zostawić w spokoju na dłużej
`ignored`	menedżer postępuje tak, jakby zasób w ogóle nie był pod opieką HA [3]	prace na maszynie bez zdejmowania jej z HA
`error`	zasób wyłączony po błędach menedżera lokalnego [3]	stan, którego nie wybieramy: wynik przekroczenia limitów

Dwa limity decydują o tym, jak uparcie menedżer ratuje maszynę, która nie chce wstać. `max_restart` to liczba prób uruchomienia na tym samym węźle, `max_relocate` to liczba prób przeniesienia na inny, a oba mają domyślnie wartość 1 [3][6]. Po ich wyczerpaniu zasób trafia w `error`, a wyjście z tego stanu wymaga najpierw przejścia w `disabled`, co opisuje dokumentacja HA (kolejność do potwierdzenia na laboratorium).

Starsze klastry używały grup HA do wskazywania węzłów. Od Proxmoksa 9.0 grupy są przestarzałe i zastąpione regułami powinowactwa: reguła węzłów (`node-affinity`) wiąże zasoby z węzłami i ich priorytetami, a reguła zasobów (`resource-affinity`) mówi, które zasoby mają działać razem albo osobno [3]. Po aktualizacji ostatniego węzła do wersji 9 grupy przekształcają się w reguły samoczynnie, a ewentualne problemy widać w dzienniku `pve-ha-crm` [3].

Reguła `resource-affinity` z `--affinity negative` rozdziela na różne węzły maszyny, które nie mogą paść razem: `dc01` i `dc02`, dwa serwery bazy z replikacją albo węzły `lb01` i `lb02`. To

odpowiednik reguł rozmieszczenia (anti-affinity) z klastra Hyper-V ze strony o Pacemakerze. Sposób, w jaki menedżer wybiera węzeł przy starcie, ustawia harmonogram zasobów klastra (CRS) w trybie basic, static albo dynamic, a opcja `ha-rebalance-on-start` pozwala użyć go także przy uruchamianiu maszyn [3][8].

Fencing: watchdog zamiast wyłączania zasilania

> **Pojęcie: watchdog.** Układ (sprzętowy albo programowy), który restartuje serwer, jeśli program nie „głaszcze” go w ustalonych odstępach. Menedżer lokalny HA odnawia go tylko wtedy, gdy węzeł ma kworum i działa prawidłowo [3].

Fencing w Proxmoksie to samoodcięcie. Węzeł, który traci kworum, nie może odnawiać watchdoga, więc po 60 sekundach watchdog go restartuje. Dopiero gdy pozostałe węzły przejmą blokadę zepsutego węzła (co dowodzi, że nie pracuje), wolno im uruchomić jego maszyny [3].

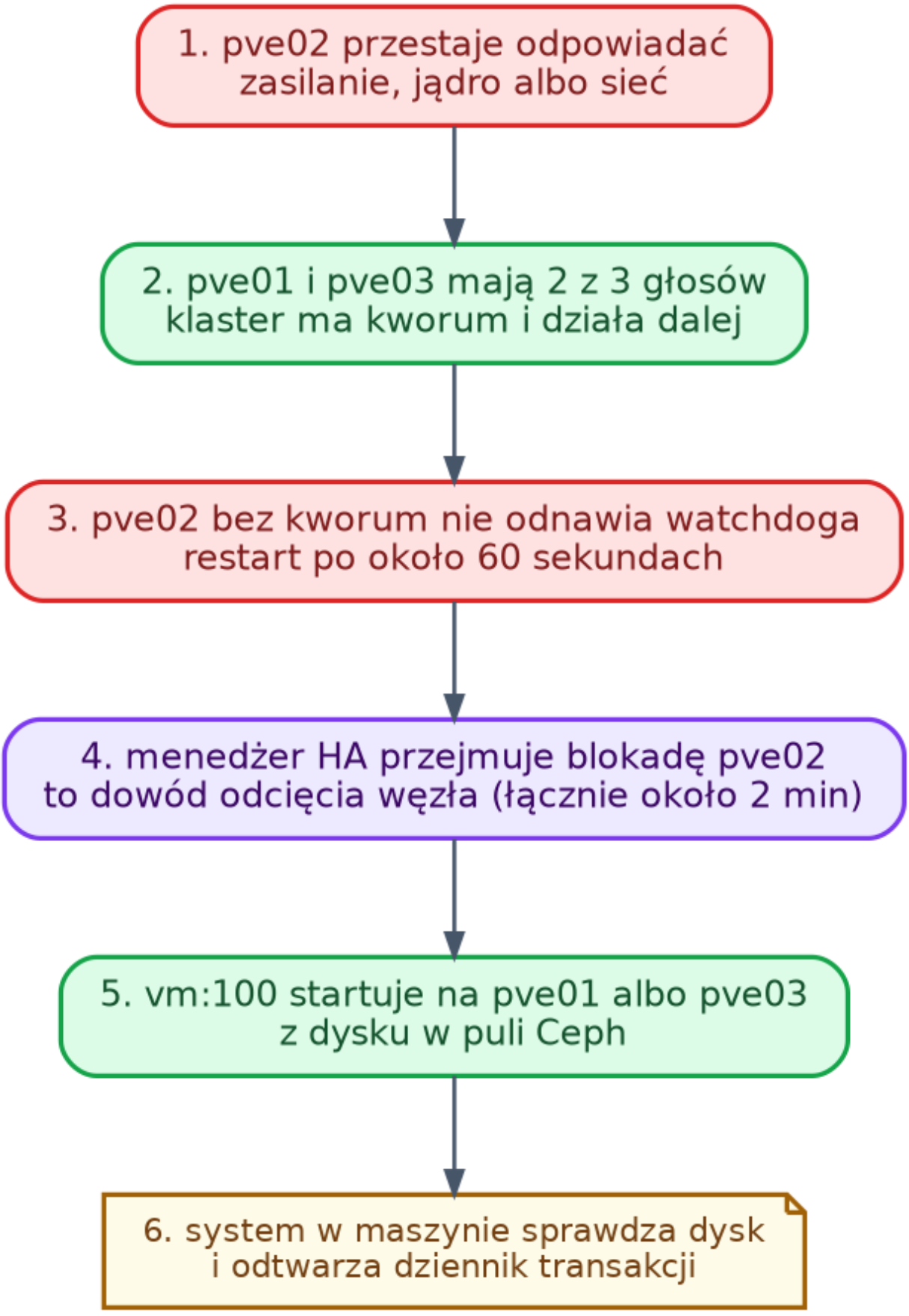
Dokumentacja HA nie opisuje wyłączania zasilania przez BMC. Strona wiki o fencingu przez IPMI , PDU i karty zarządzania dotyczy wersji 3.x i sama ostrzega, że jest nieaktualna, więc nie opieramy na niej niczego. BMC służy nam do konsoli, do testu awarii i do ręcznego sprawdzenia, że węzeł naprawdę jest wyłączony, zanim ktoś użyje `pvecm expected`, a ewentualnie jako źródło watchdoga sprzętowego (moduł `ipmi_watchdog`, dostępność i parametry do potwierdzenia w dokumentacji serwera).

Programowy `softdog` działa w jądrze tego samego węzła i według dokumentacji ma niższą niezawodność niż sprzętowy, bo nie jest niezależny od sprzętu [3]. Dokumentacja nie nakazuje sprzętowego watchdoga, ale dla produkcji go zalecamy: jeśli jądro się zawiesi, `softdog` zawiesza się razem z nim. W laboratorium wystarczy `softdog`, a przy projekcie produkcyjnym sprawdzamy, który moduł obsługuje układ naszych serwerów.

Podobna ochrona w Pacemakerze polega na odcięciu węzła przez inny węzeł (STONITH przez kartę zarządzania) i opisuje ją strona Pacemaker na Linuksie: własna usługa z adresem wirtualnym. Różnice zbieramy w tabeli na stronie Proxmox VE a Hyper-V z Failover Cluster: porównanie i przenosiny.

Co się dzieje przy awarii węzła

Dokumentacja podaje, że menedżer HA ma typowy czas wykrycia błędu i przełączenia około 2 minut i że ogranicza to dostępność do około 99,999 procent [3]. Ten czas składa się z kilku etapów, które pokazuje rysunek i tabela.


1. pve02 przestaje odpowiadać
zasilanie, jądro albo sieć
2. pve01 i pve03 mają 2 z 3 głosów
klaster ma kworum i działa dalej
3. pve02 bez kworum nie odnawia watchdoga
restart po około 60 sekundach
4. menedżer HA przejmuje blokadę pve02
to dowód odcięcia węzła (łącznie około 2 min)
5. vm:100 startuje na pve01 albo pve03
z dysku w puli Ceph
6. system w maszynie sprawdza dysk
i odtwarza dziennik transakcji

Rysunek 2. Awaria węzła pve02 z maszyną pod opieką HA. Pozostałe węzły mają 2 z 3 głosów, więc klaster działa. Węzeł bez kworum nie odnawia watchdoga, po około 60 sekundach restartuje się, a menedżer klastra przejmuje jego blokadę i uruchamia maszynę na innym węźle z danymi z puli Ceph. (kliknięcie otwiera pełny rozmiar)

Etap	Co się dzieje	Orientacyjny czas
1. Awaria	`pve02` przestaje odpowiadać: zasilanie, jądro albo sieć	0
2. Wykrycie	Corosync na `pve01` i `pve03` uznaje węzeł za nieobecny, a one mają 2 z 3 głosów	sekundy
3. Samoodcięcie	jeśli `pve02` żyje, ale nie ma kworum, nie odnawia watchdoga i restartuje się po 60 sekundach [3]	około 60 sekund
4. Przejęcie blokady	menedżer klastra przejmuje blokadę węzła, co dowodzi, że został odcięty [3]	do łącznych około 2 minut
5. Uruchomienie	maszyna `vm:100` startuje na `pve01` lub `pve03` z dysku w Ceph	zależy od systemu w maszynie

Uruchomienie po awarii jest jak start po nagłym wyłączeniu prądu: system w maszynie przechodzi kontrolę dysku i odtwarza dziennik transakcji. Dlatego aplikacje, które nie lubią twardego zatrzymania (bazy, Optima), mają własne kopie i dzienniki, a my nie traktujemy HA jako zastępstwa kopii zapasowej.

Pełny czas, od awarii do chwili, gdy użytkownik znów pracuje, mierzymy w laboratorium skryptem z sekcji „Test awarii”. Liczba z dokumentacji to punkt wyjścia, a nie obietnica dla pana Zbigniewa.

Migracja na żywo i offline

Migracja na żywo (`qm migrate --online`) kopiuje pamięć działającej maszyny na inny węzeł, a na końcu robi krótką pauzę i przełącza. Przy magazynie współdzielonym (Ceph) przesyłamy tylko pamięć, bo dyski już tam są, a bez niego użyjemy `--with-local-disks`, które kopiuje też dyski [11]. To odpowiednik migracji na żywo klastra Hyper-V ze strony Hyper-V na co dzień: przełącznik, maszyny, punkty kontrolne i migracja.

Migracja na żywo wymaga zgodnego procesora na obu węzłach. Typ procesora maszyny ustawiony na `host` przekazuje do niej wszystkie możliwości procesora węzła, więc maszyna z takim typem migruje bezpiecznie tylko między węzłami z identycznymi procesorami; dla mieszanego sprzętu wybieramy wspólny model, na przykład `x86-64-v2-AES` (do potwierdzenia

przy zakupie sprzętu). Maszyny z przekazanym urządzeniem PCI (na przykład kartą graficzną) nie migrują na żywo wcale [11].

Ruch migracji domyślnie idzie przez SSH po sieci klastra. W `datacenter.cfg` można wskazać osobną podsieć migracji (`migration: type=secure,network=CIDR`), a limity pasma ustawia `bwlimit` [8]. U nas migracja idzie siecią Ceph z VLAN 111, bo ma 10G.

Migracja offline dotyczy maszyny wyłączonej: kopiujemy dyski i konfigurację, bez pamięci. Dla maszyn na replikowanym ZFS jest szybka, bo dyski są już na węźle docelowym, w stanie z ostatniej synchronizacji [5].

Tryb konserwacji węzła

> **Pojęcie: tryb konserwacji.** Stan węzła, w którym menedżer HA przenosi z niego wszystkie zasoby i nie umieszcza na nim nowych. Włączamy go poleceniem `ha-manager crm-command node-maintenance enable NazwaWezla` [3].

Zasoby trafiają na węzły wybrane według reguł HA i trybu harmonogramu, a menedżer zapamiętuje, skąd je zabrał, więc po `disable` wszystkie wracają na swój węzeł [3]. Tryb ręczny nie znika po restarcie węzła: usuwamy go jawnie albo czyścimy stan menedżera, więc po restarcie pamiętamy o `disable` [3]. Postęp widać w `ha-manager status`, a koniec przenoszenia w dzienniku komunikatem o zamknięciu watchdoga przez menedżer lokalny [3].

Od Proxmoksa 9.2 jest też polecenie `ha-manager crm-command disarm-ha`, które wyłącza stos HA w całym klastrze (w trybie `freeze` albo `ignore`), żeby wykonać planowe prace, na przykład zmiany sieci klastra, bez fencingu. Wraca się poleceniem `arm-ha` [1][3]. Wydanie 9.2 opisuje też znany problem: wyłączony stos HA z trwającymi migracjami może zatrzymać aktualizację z wersji sprzed 9.2 [1].

Zachowanie przy zwykłym restarcie albo wyłączeniu węzła ustawia `shutdown_policy` w sekcji `ha` pliku `datacenter.cfg`. Domyślna wartość to `conditional`, która rozpoznaje, czy zażądano wyłączenia, czy restartu, a pozostałe to `freeze`, `failover` i `migrate` [8]. Do planowych prac nie polegamy na polityce, tylko włączamy tryb konserwacji.

Konfiguracja krok po kroku

Zanim zaczniemy

Potrzebujemy trzech serwerów z procesorem x86-64 z wirtualizacją (Intel VT albo AMD-V), co najmniej 2 GB pamięci na system i usługi plus pamięć dla maszyn, a dla Ceph dodatkowo pamięć na OSD [10][4]. Dokumentacja zaleca szybkie, nadmiarowe dyski (najlepiej SSD z ochroną przed utratą zasilania), a przy ZFS około 1 GB pamięci na każdy terabajt danych ponad bazowe 2 GB [10]. Serwery mają BMC w VLAN 98, a kontrolery są przestawione w tryb HBA, według strony o RAID i kontrolerach.

Host	Adres w VLAN 10	Adres w VLAN 101 (link0)	Adres w VLAN 111 (Ceph)	BMC w VLAN 98
`pve01`	`10.20.10.90`	`10.20.101.90`	`10.20.111.90`	`10.20.98.90`
`pve02`	`10.20.10.91`	`10.20.101.91`	`10.20.111.91`	`10.20.98.91`
`pve03`	`10.20.10.92`	`10.20.101.92`	`10.20.111.92`	`10.20.98.92`
`pbs01` (kopie, osobna strona)	`10.20.10.93`	brak	brak	`10.20.98.93`
`pbs02` (drugi PBS, propozycja)	`10.20.10.94`	brak	brak	`10.20.98.94`

Sprawdziliśmy grep po całej bazie, że adresy od `.90` do `.94` w sieciach `10.20.10.0/24` i `10.20.98.0/24` są wolne, tak samo jak podsieci `10.20.101.0/24` i `10.20.111.0/24`. Nazwy `pve01.ad.zielona.example` i pozostałe wpisujemy do DNS na `dc01` i do NetBoksa, a zegary synchronizujemy z kontrolerów domeny: Proxmox używa domyślnie programu `chrony`, a serwery czasu wskazujemy w `/etc/chrony/chrony.conf` [7].

Krok 1. Instalacja Proxmoksa na każdym węźle

Obraz ISO instalatora pobieramy ze strony producenta i uruchamiamy z wirtualnego nośnika BMC (strona Zdalne zarządzanie: BMC, IPMI i Redfish w osobnej sieci). Instalator oferuje ext4, XFS, ZFS (z programowym RAID) i Btrfs jako podgląd technologii [10]. Dla systemu wybieramy ZFS RAID1 na dwóch małych dyskach przy kontrolerze w trybie HBA albo ext4 na lustrze z kontrolera, ale nie ZFS na sprzętowym RAID, bo dokumentacja mówi wprost, że to nie jest wspierane i grozi utratą danych [10].

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ekran instalatora Proxmox VE z wyborem dysku docelowego (Options, ZFS RAID1 z dwoma małymi dyskami zaznaczonymi), a obok ekran z adresem `10.20.10.90/24`, bramą `10.20.10.1` i nazwą `pve01.ad.zielona.example`.

Po instalacji wchodzimy na `https://10.20.10.90:8006` jako root. Repozytorium korporacyjne działa tylko z kluczem subskrypcji, a o repozytoriach i aktualizacjach piszemy na stronie Proxmox VE w utrzymaniu: kopie PBS, aktualizacje, użytkownicy.

Krok 2. Sieć węzła

Konfigurację sieci trzymamy w `/etc/network/interfaces`. Poniższy plik dla `pve01` łączy trzy sieci: osobny port 1 Gb na Corosync, dwa łącza 10G w `bond0` pod mostem świadomym VLAN-ów oraz adresy zarządzania i Ceph jako interfejsy VLAN na moście [7]. Nazwy kart (`eno1`, `enp65s0f0`, `enp65s0f1`) są przykładowe, a własne odczytujemy z `ip -br link`.

Pokaż pełny przykład (36 linii)

```
auto lo
iface lo inet loopback

# Corosync link0: osobny port, osobna sieć, bez bramy
auto eno1
iface eno1 inet static
        address 10.20.101.90/24

iface enp65s0f0 inet manual
iface enp65s0f1 inet manual

auto bond0
iface bond0 inet manual
        bond-slaves enp65s0f0 enp65s0f1
        bond-miimon 100
        bond-mode active-backup
        bond-primary enp65s0f0

auto vmbr0
iface vmbr0 inet manual
        bridge-ports bond0
        bridge-stp off
        bridge-fd 0
        bridge-vlan-aware yes
        bridge-vids 2-4094
```

```
# Zarządzanie (VLAN 10), Corosync link1
auto vmbr0.10
iface vmbr0.10 inet static
        address 10.20.10.90/24
        gateway 10.20.10.1

# Ceph (VLAN 111), bez bramy
auto vmbr0.111
iface vmbr0.111 inet static
        address 10.20.111.90/24
```

Na `pve02` i `pve03` zmieniamy tylko trzy adresy (`.91` i `.92`). Po zapisaniu pliku wykonujemy `ifreload -a` z konsoli BMC, a nie przez SSH. Maszynie przypisujemy sieć i VLAN na jej karcie wirtualnej: `qm set NumerMaszyny --net0 virtio,bridge=vmbr0,tag=10` [7].

Krok 3. Klaster

Na `pve01` tworzymy klaster z dwoma łączami (adresy z tabeli):

```
pvecm create zielona-pve --link0 10.20.101.90,priority=20 --link1 10.20.10.90,priority=10
```

Na `pve02` dołączamy węzeł (i analogicznie na `pve03` z adresami `.92`):

```
pvecm add 10.20.10.90 --link0 10.20.101.91 --link1 10.20.10.91
```

Sprawdzamy `pvecm status` (oczekujemy trzech węzłów i `Quorate: Yes`) oraz `corosync-cfgtool -s` (oba łącza połączone do obu pozostałych węzłów). Reguły zapory muszą przepuszczać UDP 5405 do 5412 między węzłami i TCP 22 [2].

Krok 4. Ceph

Na każdym węźle uruchamiamy `pveceph install`. Potem na jednym `pveceph init --network 10.20.111.0/24`, a na każdym węźle `pveceph mon create` i `pveceph mgr create`. Dyski dodajemy poleceniem `pveceph osd create` po jednym na dysk, a po dodaniu wszystkich tworzymy pulę `pveceph pool create vm-pula --add_storages` [4]. Stan sprawdzamy `ceph -s`: oczekujemy `HEALTH_OK`, trzech monitorów w kworum i wszystkich OSD w stanie `up` i `in`.

ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ekran Datacenter, Ceph w interfejsie WWW po utworzeniu puli, widoczne zdrowie `HEALTH_OK`, trzy monitory, sześć OSD i pula `vm-pula`.

Krok 5. Maszyna testowa pod opieką HA

Tworzymy maszynę testową `ci-test` z numerem 100 na magazynie `vm-pula`, instalujemy w niej agenta gościa (`qemu-guest-agent`) i dodajemy ją do HA:

```
ha-manager add vm:100 --state started
ha-manager rules add node-affinity ci-test-wezly --resources vm:100 --nodes
pve01:2,pve02:1,pve03:1
ha-manager status
```

Jeśli maszyna ma stać na replikowanym ZFS zamiast Ceph, dodajemy jeszcze `pvesr create-local-job 100-0 pve02 --schedule "*/15"` i drugie zadanie na `pve03`.

Krok 6. Test awarii

Przed testem uruchamiamy z innego komputera (na przykład z `mon01` albo ze stacji administratora) skrypt, który co sekundę pyta maszynę i zapisuje, kiedy przestała odpowiadać i kiedy wróciła.

```
#!/usr/bin/env bash
# Pomiar przerwy w odpowiedzi maszyny: pytamy co sekundę i zapisujemy zmiany stanu.
AdresMaszyny="${1:?podaj adres maszyny}"
stan="start"
poczatek=""
while true; do
  if ping -c 1 -W 1 "$AdresMaszyny" > /dev/null 2>&1; then
    nowy="gora"
  else
    nowy="dol"
  fi
  if [ "$nowy" != "$stan" ]; then
    teraz=$(date +%T)
    if [ "$nowy" = "dol" ]; then
      poczatek=$(date +%s)
      echo "$teraz maszyna przestala odpowiadac"
```

```
    elif [ -n "$poczatek" ]; then
      echo "$teraz maszyna wrocila, przerwa: $(( $(date +%s) - poczatek )) s"
    else
      echo "$teraz maszyna odpowiada"
    fi
    stan="$nowy"
  fi
  sleep 1
done
```

Następnie wyłączamy zasilanie węzła, na którym stoi `vm:100`, poleceniem `ipmitool chassis power off` przez BMC. W `ha-manager status` obserwujemy kolejne etapy, a skrypt zapisuje całkowitą przerwę. Wynik zapisujemy na stronie usługi, razem z liczbą z dokumentacji (około 2 minuty).

> Test awarii wykonujemy tylko na maszynach laboratoryjnych. Przed testem sprawdzamy, że `ceph -s` pokazuje `HEALTH_OK`, że w klastrze nie trwa żadna migracja i że mamy konsolę BMC pozostałych węzłów, bo przy błędzie w sieci Corosync oba pozostałe węzły mogą też stracić kworum.

Wycofanie zmian

Maszynę zdejmujemy z HA poleceniem `ha-manager remove vm:100`, a w razie potrzeby wracamy do zwykłego uruchamiania ręcznego. Węzeł usuwamy z klastra tak: wyłączamy go, na pozostałych wykonujemy `pvecm delnode NazwaWezla`, a przed ponownym użyciem tego samego serwera instalujemy go od nowa, bo stara konfiguracja klastra nie może wrócić na ten sam adres i pod tą samą nazwą [2]. Ceph rozbieramy w odwrotnej kolejności: OSD, potem monitory i menedżery, na końcu pule.

Częste błędy

Objaw	Przyczyna	Co robimy
`Quorate: No` po restarcie jednego z dwóch węzłów	klaster dwuwęzłowy bez urządzenia kworum	dokładamy QDevice na hoście spoza klastra albo trzeci węzeł
Węzeł nie dołącza do klastra	niezgodne wersje, ślad po starej konfiguracji, zapora blokuje UDP 5405 do 5412	porównujemy `pveversion`, czyścimy węzeł, sprawdzamy zaporę [2]
`HEALTH_WARN` po awarii węzła w klastrze z trzema węzłami	nie ma czwartego węzła, na którym Ceph odtworzy trzecią kopię	stan jest oczekiwany do powrotu węzła, a drugiej awarii nie wolno dopuścić

Objaw	Przyczyna	Co robimy
Wszystkie maszyny stoją po awarii drugiego węzła	liczba kopii spadła poniżej `min_size`	przywracamy węzeł, nie obniżamy `min_size` do 1 [4]
Maszyna nie wstaje na innym węźle	maszyna nie jest zasobem HA albo stoi na lokalnym magazynie bez replikacji	`ha-manager config`, przeniesienie dysków do Ceph albo replikacja ZFS
Maszyna w stanie `error`	wyczerpane `max_restart` i `max_relocate`	usuwamy przyczynę, ustawiamy `disabled`, potem `started` (kolejność do potwierdzenia) [3]
Migracja na żywo kończy się błędem	procesor typu `host` na różnych węzłach albo urządzenie przypięte do węzła	wspólny model procesora, odpięcie urządzeń [11]
Węzeł restartuje się sam przy zerwaniu sieci	prawidłowe samoodcięcie bez kworum	sprawdzamy `corosync-cfgtool -s` i kable link0 oraz link1
Po zmianie sieci węzeł znika z klastra	zmiana adresu Corosync	adresów i nazw po utworzeniu klastra nie zmieniamy [2]
Po restarcie węzła zasoby nie wracają	tryb konserwacji nadal włączony	`ha-manager crm-command node-maintenance disable NazwaWezla` [3]
Interfejs WWW przestaje zapisywać zmiany	klaster bez kworum, `/etc/pve` tylko do odczytu	przywracamy kworum, zamiast obchodzić je poleceniem `pvecm expected`

□□ Ćwiczenia

1. Dlaczego klaster z trzema węzłami nie potrzebuje urządzenia kworum, a z dwoma tak?

Odpowiedź

Trzy węzły po jednym głosie dają większość z dwóch, więc po awarii jednego dwa pozostałe nadal mają kworum. Dwa węzły mają po jednym głosie z dwóch, a utrata jednego zostawia połowę, czyli za mało. QDevice dokłada trzeci głos spoza klastra, więc po awarii węzła zostają dwa z trzech.

1. Pula Ceph ma `size` 3 i `min_size` 2 na trzech węzłach. Co się dzieje po awarii jednego węzła i co po awarii drugiego?

Odpowiedź

Po awarii jednego zostają dwie kopie, czyli tyle, ile wynosi `min_size`: maszyny działają, a klaster pokazuje `HEALTH_WARN`, bo nie ma gdzie odtworzyć trzeciej kopii. Po awarii drugiego zostaje

jedna kopia, poniżej `min_size`, więc Ceph przestaje obsługiwać operacje na tych danych i maszyny stają.

1. Co tracimy przy awarii węzła, gdy maszyna stoi na ZFS z replikacją co 15 minut, a co przy Ceph?

Odpowiedź

Przy replikacji ZFS tracimy zmiany od ostatniej udanej replikacji, czyli do około 15 minut. Przy Ceph tracimy tylko to, co aplikacja miała w pamięci, bo zapis jest potwierdzany dopiero po zapisie kopii na wielu węzłach, więc dyski są aktualne.

1. Dlaczego węzeł, który stracił kworum, restartuje się sam i czemu to dobrze?

Odpowiedź

Bez kworum menedżer lokalny nie może odnawiać watchdoga, więc po około 60 sekundach watchdog restartuje węzeł. Dzięki temu pozostałe węzły mają pewność, że jego maszyny przestały działać, zanim uruchomią je u siebie, i żadna maszyna nie pracuje w dwóch miejscach naraz.

1. Co robimy przed aktualizacją i restartem węzła, który prowadzi maszyny pod opieką HA?

Odpowiedź

Włączamy tryb konserwacji poleceniem `ha-manager crm-command node-maintenance enable NazwaWezla`, czekamy, aż zasoby przejdą na inne węzły, wykonujemy prace i restart, a po powrocie węzła wyłączamy tryb poleceniem z `disable`. Zasoby wracają wtedy na swój węzeł, a o `disable` trzeba pamiętać, bo tryb nie znika po restarcie.

Powiązane: Wysoka dostępność: usługa, która nie ma jednego serwera · Wysoka dostępność: pojęcia, liczenie dostępności i poziomy rezerwy · Replikacja danych między hostami: DRBD, GlusterFS, Ceph, ZFS i replikacja w aplikacji · Klaster Windows dla własnych usług: rola, zasób, zależności, kworum i testy przełączenia · Pacemaker na Linuksie: własna usługa z adresem wirtualnym · Macierz wysokiej dostępności usług Zielonej: poziom rezerwy każdej usługi, zależności i kolejność prac · Proxmox VE w utrzymaniu: kopie PBS, aktualizacje, użytkownicy · Proxmox VE a Hyper-V z Failover Cluster: porównanie i przenosiny · RAID i kontrolery: sprzętowy, programowy i HBA dla S2D · Zdalne zarządzanie: BMC, IPMI i Redfish w osobnej sieci · Przełączniki: L2 i L3, tablica

MAC, VLAN, LACP i STP · Hyper-V na co dzień: przełącznik, maszyny, punkty kontrolne i migracja

Źródła: (odczyt 3 października 2026)

1. pve.proxmox.com: Roadmap (wydania 9.1 z 19 listopada 2025 i 9.2 z 21 maja 2026, Debian 13 „Trixie”, jądro 7.0, QEMU 11.0, ZFS 2.4, Ceph Squid 19.2.3 i Tentacle 20.2.1; wyłączanie HA w 9.2)
2. pve.proxmox.com: Cluster Manager (wymagania sieci, `pvecm`, łącza, QDevice)
3. pve.proxmox.com: High Availability (wymagania, stany, reguły, watchdog, czas przełączenia, tryb konserwacji, `disarm-ha`)
4. pve.proxmox.com: Deploy Hyper-Converged Ceph Cluster (`pveceph`, wymagania sprzętowe, `size` i `min_size`)
5. pve.proxmox.com: Storage Replication i pvesr(1)
6. pve.proxmox.com: ha-manager(1)
7. pve.proxmox.com: Host System Administration (sieć, mosty, VLAN, agregacja, czas)
8. pve.proxmox.com: datacenter.cfg(5) (`shutdown_policy`, CRS, sieć migracji)
9. pve.proxmox.com: Storage (rodzaje magazynów, współdzielenie)
10. pve.proxmox.com: Installation, System Requirements i Installation (wymagania, systemy plików, licencja AGPLv3)
11. pve.proxmox.com: qm(1) i Qemu/KVM Virtual Machines (migracja, `--with-local-disks`)

Revision #12
Created 2026-10-03 19:11:13 UTC by Tomasz Konieczny
Updated 2026-10-26 14:21:35 UTC by Tomasz Konieczny